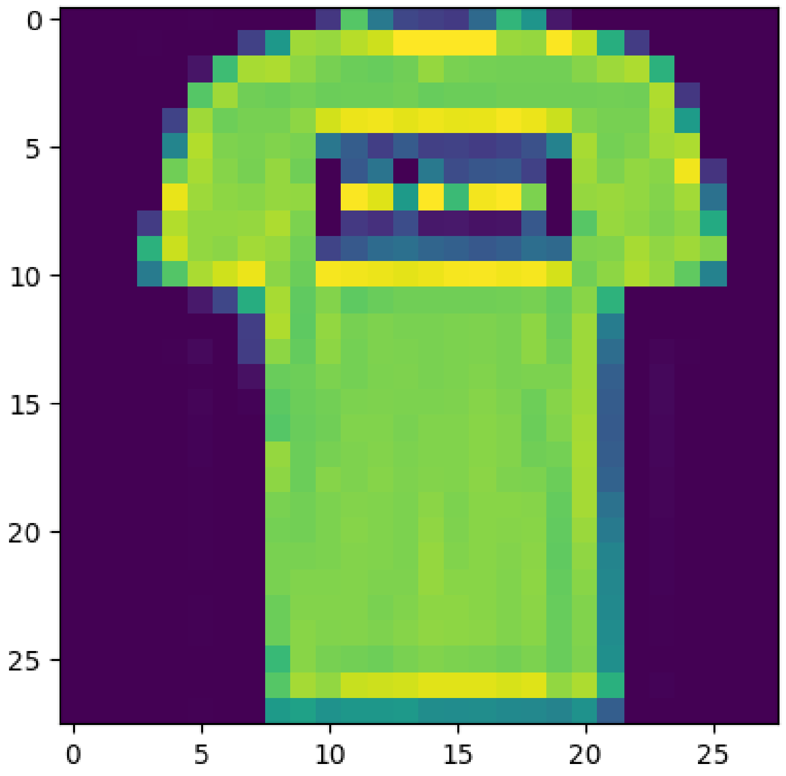
```python
import tensorflow as tf
import matplotlib.pyplot as plt
from tensorflow import keras
import numpy as np
```

```python
(x_train, y_train), (x_test, y_test) =
keras.datasets.fashion_mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-
keras-datasets/train-labels-idx1-ubyte.gz
29515/29515 ━━━━━━━━━━━━━━━━━━━━ 0s 2us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-
keras-datasets/train-images-idx3-ubyte.gz
26421880/26421880 ━━━━━━━━━━━━━━━━━━━━ 19s 1us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-
keras-datasets/t10k-labels-idx1-ubyte.gz
5148/5148 ━━━━━━━━━━━━━━━━━━━━ 0s 1us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-
keras-datasets/t10k-images-idx3-ubyte.gz
4422102/4422102 ━━━━━━━━━━━━━━━━━━━━ 2s 0us/step
```

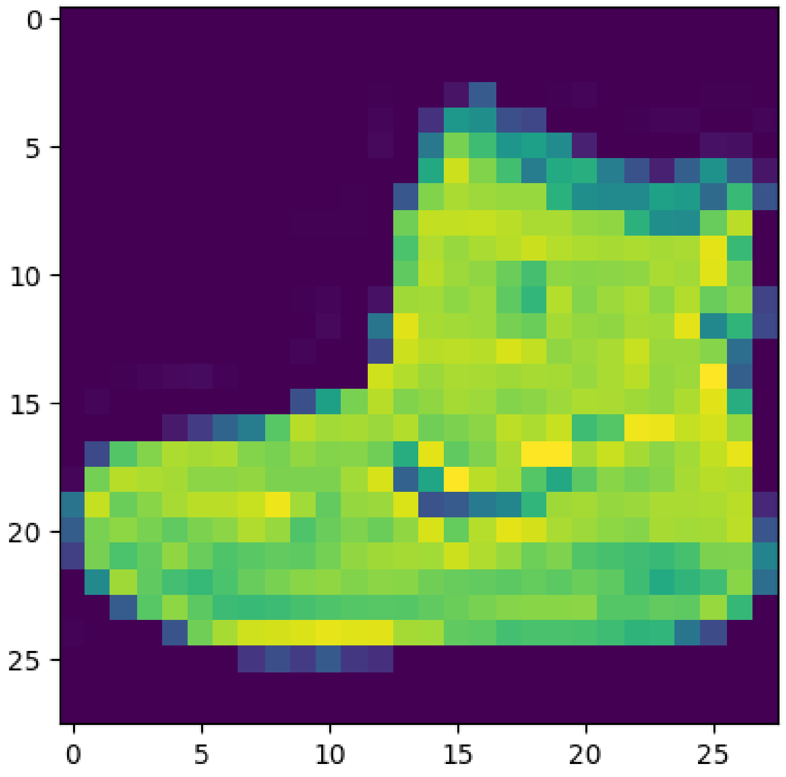
```python
plt.imshow(x_train[1])
```

```
<matplotlib.image.AxesImage at 0x1d60abfdc70>
```



```
plt.imshow(x_train[0])
```

```
<matplotlib.image.AxesImage at 0x1d60b32cc20>
```



```
x_train = x_train.astype('float32') / 255.0
x_test = x_test.astype('float32') / 255.0
x_train = x_train.reshape(-1, 28, 28, 1)
x_test = x_test.reshape(-1, 28, 28, 1)
```

```
x_train.shape
```

```
(60000, 28, 28, 1)
```

```
x_test.shape
```

```
(10000, 28, 28, 1)
```

```
y_train.shape
```

```
(60000,)
```

```
y_test.shape
```

```
(10000,)
```

```
model = keras.Sequential([
keras.layers.Conv2D(32, (3,3), activation='relu',
input_shape=(28,28,1)),


keras.layers.MaxPooling2D((2,2)),

keras.layers.Dropout(0.25),

keras.layers.Conv2D(64,  (3,3),  activation='relu'),


keras.layers.MaxPooling2D((2,2)),

keras.layers.Dropout(0.25),
keras.layers.Conv2D(128, (3,3), activation='relu'),



keras.layers.Flatten(),
keras.layers.Dense(128, activation='relu'),


keras.layers.Dropout(0.25),
keras.layers.Dense(10, activation='softmax')

])
```

```
C:\Users\user\AppData\Local\Programs\Python\Python312\Lib\site-
packages\keras\src\layers\convolutional\base_conv.py:107: UserWarning:
Do not pass an `input_shape`/`input_dim` argument to a layer. When
using Sequential models, prefer using an `Input(shape)` object as the
first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer,
**kwargs)
```

```
model.summary()
```

```
Model: "sequential"

┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳
━━━━━━━━━━━━━━━━━┓
┃ Layer (type)                          ┃ Output Shape                 ┃
Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇
━━━━━━━━━━━━━━━━━┩
│ conv2d (Conv2D)                       │ (None, 26, 26, 32)           │
320 │
├───────────────────────────────────────┼──────────────────────────────┼
─────────────────┤
│ max_pooling2d (MaxPooling2D)          │ (None, 13, 13, 32)           │
```

```
0 │
├─────────────────────────────────────┼───────────────────────────┼
────────────────┤
│ dropout (Dropout)                   │ (None, 13, 13, 32)        │
0 │
├─────────────────────────────────────┼───────────────────────────┼
────────────────┤
│ conv2d_1 (Conv2D)                   │ (None, 11, 11, 64)        │
18,496 │
├─────────────────────────────────────┼───────────────────────────┼
────────────────┤
│ max_pooling2d_1 (MaxPooling2D)      │ (None, 5, 5, 64)          │
0 │
├─────────────────────────────────────┼───────────────────────────┼
────────────────┤
│ dropout_1 (Dropout)                 │ (None, 5, 5, 64)          │
0 │
├─────────────────────────────────────┼───────────────────────────┼
────────────────┤
│ conv2d_2 (Conv2D)                   │ (None, 3, 3, 128)         │
73,856 │
├─────────────────────────────────────┼───────────────────────────┼
────────────────┤
│ flatten (Flatten)                   │ (None, 1152)              │
0 │
├─────────────────────────────────────┼───────────────────────────┼
────────────────┤
│ dense (Dense)                       │ (None, 128)               │
147,584 │
├─────────────────────────────────────┼───────────────────────────┼
────────────────┤
│ dropout_2 (Dropout)                 │ (None, 128)               │
0 │
├─────────────────────────────────────┼───────────────────────────┼
────────────────┤
│ dense_1 (Dense)                     │ (None, 10)                │
1,290 │
└─────────────────────────────────────┴───────────────────────────┴
────────────────┘

 Total params: 241,546 (943.54 KB)

 Trainable params: 241,546 (943.54 KB)

 Non-trainable params: 0 (0.00 B)

model.compile(optimizer='adam',
loss='sparse_categorical_crossentropy', metrics=['accuracy'])
history =
model.fit(x_train,y_train,epochs=10,validation_data=(x_test,y_test))
```

```
Epoch 1/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 38s 19ms/step - accuracy: 0.7072 -
loss: 0.7836 - val_accuracy: 0.8525 - val_loss: 0.3990
Epoch 2/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 37s 17ms/step - accuracy: 0.8596 -
loss: 0.3868 - val_accuracy: 0.8847 - val_loss: 0.3160
Epoch 3/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 32s 17ms/step - accuracy: 0.8789 -
loss: 0.3313 - val_accuracy: 0.8933 - val_loss: 0.2881
Epoch 4/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 33s 17ms/step - accuracy: 0.8891 -
loss: 0.3005 - val_accuracy: 0.8905 - val_loss: 0.2983
Epoch 5/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 33s 17ms/step - accuracy: 0.8976 -
loss: 0.2776 - val_accuracy: 0.9007 - val_loss: 0.2650
Epoch 6/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 33s 18ms/step - accuracy: 0.9005 -
loss: 0.2666 - val_accuracy: 0.9047 - val_loss: 0.2543
Epoch 7/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 33s 18ms/step - accuracy: 0.9077 -
loss: 0.2462 - val_accuracy: 0.9037 - val_loss: 0.2700
Epoch 8/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 34s 18ms/step - accuracy: 0.9102 -
loss: 0.2418 - val_accuracy: 0.9098 - val_loss: 0.2503
Epoch 9/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 38s 20ms/step - accuracy: 0.9137 -
loss: 0.2321 - val_accuracy: 0.9051 - val_loss: 0.2521
Epoch 10/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 35s 19ms/step - accuracy: 0.9164 -
loss: 0.2226 - val_accuracy: 0.9071 - val_loss: 0.2568
```

```
test_loss,test_acc = model.evaluate(x_test,y_test)
print('Test accuracy',test_acc)
```

```
313/313 ━━━━━━━━━━━━━━━━━━━━ 2s 7ms/step - accuracy: 0.9055 - loss:
0.2724
Test accuracy 0.9071000218391418
```